

# QtoPSurDistance.txt

功能：求曲面 $q(u, v)$ 上一点 $q(u_0, v_0) = (q_x, q_y, q_z)$ 到曲面 $p(u, v)$ 距离。

输入参数： $(q_x, q_y)$ —曲面 $q(u, v)$ 上一点； $(u_0, v_0)$ — $q(u, v)$ 上一点 $(q_x, q_y)$ 的一对参数。定义曲面 $q(u, v)$ 的控制顶点 $m\_newVertex$ 与节点矢量 $m\_newNode$ 及定义曲面 $p(u, v)$ 的控制顶点 $m\_aVertex$ 与节点矢量 $m\_aNode$ 均是受保护成员。

输出参数： $d$ —曲面 $q(u, v)$ 上一点 $q(u_0, v_0) = (q_x, q_y, q_z)$ 到曲面 $p(u, v)$ 距离； $(p_x, p_y, p_z)$ —垂足或点在曲面上的投影。

调用函数：GetSurDerivat—求曲面 $p(u, v)$ 上的点与偏导矢；dsense—梯度法解非线性方程组，见文献[79]。

```
void QtoPSurDistance(double u0, double v0, double &qx, double &qy, double &qz,
                    double &px, double &py, double &pz, double &d)
{ /* 这里以q(u, v)上一点 (qx, qy, qz)的一对参数 (u0, v0)为其在曲面p(u, v)上投影点的参数初值 */
    int i;
    CArray<double, double> x;
    x.SetSize(2);
    double up, vp;
    GetSurDerivat(0, 0, u0, v0, px, py, pz);
    d=sqrt((px-qx)*(px-qx)+(py-qy)*(py-qy)+(pz-qz)*(pz-qz));
    if(d==0) return;
    x[0]=u0; x[1]=v0;
    int js=1000;
    double eps=0.000000000001;
label: i=dsense(eps, qx, qy, qz, x, js); /* 解非线性方程组，若i=-1表示遇到分母D=0，目标
                                         函数出现决不计值点；若i=0表示未进入迭代，初值即是解；
                                         i=js为已迭代了最大迭代次数，仍未满足精度要求 */
    if(i==-1) /* 处理出现D=0的情况，改变初值*/
    {
        x[0]=x[0]+0.001; x[1]=x[1]+0.001; goto label;
    }
    up=x[0]; vp=x[1]; /* 得曲面参数up, vp */
    GetSurDerivat(0, 0, up, vp, px, py, pz); /* 求垂足(px, py, pz) */
    d=sqrt((px-qx)*(px-qx)+(py-qy)*(py-qy)+(pz-qz)*(pz-qz)); /* 点q到曲面距离d */
}
```